**Coding interview study plan - what to study and practice based on time left**

One of the most important questions to answer at the start of your coding interview preparation is: What study topics and practice questions should you do to most efficiently prepare for your coding interviews?

There are plenty of resources on the internet, but it can be hard to know how they fit into the time you have left to prepare. Thankfully, this article will help you with that.

I have personally gone through the dreaded Software Engineer interview process myself several times and prepared my own study plans, refining them each time.

In this article, I will share the 3 month study plan that I personally use to prepare for my coding interviews. You will find the exact topics to study (with recommended links) and exact questions to practice (with practice links).

**Recommended preparation time and approach**[**​**](https://www.techinterviewhandbook.org/coding-interview-study-plan/#recommended-preparation-time-and-approach)

How much time do you need to prepare for your coding interviews? Generally, 3 months (if you can dedicate 11 hours a week) is the recommended period of time for a more holistic preparation. I will be sharing recommended study plans for 3 months (recommended period), but you can generate study plans for practice questions for any time frame you need via the Grind 75 tool (built by me). More options like filtering by difficulty, topics, alternative grouping of questions can be found there.

Regardless of how long you have, if you are unfamiliar with core data structures and algorithms knowledge, you are advised to revise them before starting on the coding interview questions practice. Different people have different styles of practicing and you should do what works best for you. The various possible approaches are:

1. **Breadth-first preparation** - Revise every topic and then start practicing a variety of questions across all topics. This strategy is recommended if you have around a month to spare.
2. **Depth-first preparation** - Tackle one topic at a time - revise materials for a topic, practice lots of questions for that topic. After ensuring mastery of a topic, move on to the next topic. Repeat for all or selected topics. If you don't have much time, this might be the best way to prepare. You can focus on the High priority topics in our recommended study plan.
3. **Depth-first-then-breadth preparation** - Tackle one topic at a time - revise materials for a topic, practice a few questions for that topic. After ensuring mastery, move on to the next topic. Repeat for all topics. At the end, practice a variety of questions across all topics. This strategy takes more time than others, so it's recommended if you have more than a month.

My personal recommendation would be the **Breadth-first preparation** or **Depth-first-then-breadth preparation**. It's important to have some form of breadth-level studying / practicing in your schedule so that you don't forget about the earlier topics as you move on to later topics.

**[Craft the perfect resume for Google and Facebook](https://www.faangtechleads.com/?utm_source=techinterviewhandbook&utm_medium=referral&utm_content=in_doc&aff=1e80c401fe7e2" \t "_blank)**[Save time crafting your resume with FAANG Tech Leads' FAANG-quality resume templates and examples which have helped many Software Engineers get interviews at top Bay Area companies!](https://www.faangtechleads.com/?utm_source=techinterviewhandbook&utm_medium=referral&utm_content=in_doc&aff=1e80c401fe7e2" \t "_blank)

**The 3 month study plan - with recommended study resources and practice question links**[**​**](https://www.techinterviewhandbook.org/coding-interview-study-plan/#the-3-month-study-plan---with-recommended-study-resources-and-practice-question-links)

In each study plan, you will find a list of coding interview topics with resources and practice questions that you should work through **in order of priority** every week.

To best utilize it, you should create a template where you break down the dates left and hours left per day, so that you can later fill in the topics/questions to cover per day.

Keep the estimate relatively conservative so you don't end up burning out.

**Week 1 - 4: Topical study + practice**[**​**](https://www.techinterviewhandbook.org/coding-interview-study-plan/#week-1---4-topical-study--practice)

These are all the topics you should study, in order of priority. The learning resources linked are my algorithm cheatsheets - which give you an overview of must-remembers like time complexity, corner cases, and topic-specific useful techniques, as well as essential and recommended practice questions.

Don't forget to apply behaviors from [coding interview best practices](https://www.techinterviewhandbook.org/coding-interview-cheatsheet/) and methods from [coding interview techniques](https://www.techinterviewhandbook.org/coding-interview-techniques/) early on while you practice!

**Week 1**[**​**](https://www.techinterviewhandbook.org/coding-interview-study-plan/#week-1)

| **Topic** | **Priority** | **Time required** |
| --- | --- | --- |
| [Array](https://www.techinterviewhandbook.org/algorithms/array/) | High | 2 hours |
| [String](https://www.techinterviewhandbook.org/algorithms/string/) | High | 3 hours |
| [Hash Table](https://www.techinterviewhandbook.org/algorithms/hash-table/) | Mid | 3 hours |
| [Recursion](https://www.techinterviewhandbook.org/algorithms/recursion/) | Mid | 3 hours |

**Week 2**[**​**](https://www.techinterviewhandbook.org/coding-interview-study-plan/#week-2)

| **Topic** | **Priority** | **Time required** |
| --- | --- | --- |
| [Sorting and searching](https://www.techinterviewhandbook.org/algorithms/sorting-searching/) | High | 3 hours |
| [Matrix](https://www.techinterviewhandbook.org/algorithms/matrix/) | High | 1 hour |
| [Linked List](https://www.techinterviewhandbook.org/algorithms/linked-list/) | Mid | 3 hours |
| [Queue](https://www.techinterviewhandbook.org/algorithms/queue/) | Mid | 2 hours |
| [Stack](https://www.techinterviewhandbook.org/algorithms/stack/) | Mid | 2 hours |

**Week 3**[**​**](https://www.techinterviewhandbook.org/coding-interview-study-plan/#week-3)

| **Topic** | **Priority** | **Time required** |
| --- | --- | --- |
| [Tree](https://www.techinterviewhandbook.org/algorithms/tree/) | High | 4 hours |
| [Graph](https://www.techinterviewhandbook.org/algorithms/graph/) | High | 4 hours |
| [Heap](https://www.techinterviewhandbook.org/algorithms/heap/) | Mid | 3 hours |
| [Trie](https://www.techinterviewhandbook.org/algorithms/trie/) | Mid | 3 hours |

**Week 4**[**​**](https://www.techinterviewhandbook.org/coding-interview-study-plan/#week-4)

| **Topic** | **Priority** | **Time required** |
| --- | --- | --- |
| [Interval](https://www.techinterviewhandbook.org/algorithms/interval/) | Mid | 2 hours |
| [Dynamic programming](https://www.techinterviewhandbook.org/algorithms/dynamic-programming/) | Low | 4 hours |
| [Binary](https://www.techinterviewhandbook.org/algorithms/binary/) | Low | 2 hours |
| [Math](https://www.techinterviewhandbook.org/algorithms/math/) | Low | 1 hour |
| [Geometry](https://www.techinterviewhandbook.org/algorithms/geometry/) | Low | 1 hour |

**[Stop grinding LeetCode. Study with a plan](https://shareasale.com/r.cfm?b=1873647&u=3114753&m=114505&urllink=&afftrack=" \t "_blank)**[Developed by Google engineers, AlgoMonster is the fastest way to get a software engineering job. Check it out for free!](https://shareasale.com/r.cfm?b=1873647&u=3114753&m=114505&urllink=&afftrack=" \t "_blank)

**Week 5 - 12: In-depth practice**[**​**](https://www.techinterviewhandbook.org/coding-interview-study-plan/#week-5---12-in-depth-practice)

Here, I listed 75 questions that you should do to be fully prepared for your coding interviews. This list of questions are generated from the [**Grind 75 tool**](https://www.techinterviewhandbook.org/grind75/) (built by me), which generates recommended study plans for coding interviews based on the time you have left. More options like filtering by difficulty, topics, alternative grouping of questions can be found there.

* If you followed the Week 1 - 4 study plan, you would have done some of these questions here. Feel free to skip them or do them again.
* Feel free to skip the dynamic programming questions if you haven't studied them or feel that they won't be relevant. Many dynamic programming questions can be solved with recursion / backtracking anyway.

Don't forget to apply behaviors from [coding interview best practices](https://www.techinterviewhandbook.org/coding-interview-cheatsheet/) and methods from [coding interview techniques](https://www.techinterviewhandbook.org/coding-interview-techniques/) early on while you practice!

We recommend using the [**Grind 75**](https://www.techinterviewhandbook.org/grind75/) tool which allows you to keep track of your practice progress.

**Week 5**

| **Problem** | **Difficulty** | **Duration** |
| --- | --- | --- |
| [Two Sum](https://leetcode.com/problems/two-sum) | **Easy** | 15 mins |
| [Valid Parentheses](https://leetcode.com/problems/valid-parentheses) | **Easy** | 20 mins |
| [Merge Two Sorted Lists](https://leetcode.com/problems/merge-two-sorted-lists) | **Easy** | 20 mins |
| [Best Time to Buy and Sell Stock](https://leetcode.com/problems/best-time-to-buy-and-sell-stock) | **Easy** | 20 mins |
| [Valid Palindrome](https://leetcode.com/problems/valid-palindrome) | **Easy** | 15 mins |
| [Invert Binary Tree](https://leetcode.com/problems/invert-binary-tree) | **Easy** | 15 mins |
| [Valid Anagram](https://leetcode.com/problems/valid-anagram) | **Easy** | 15 mins |
| [Binary Search](https://leetcode.com/problems/binary-search) | **Easy** | 15 mins |
| [Flood Fill](https://leetcode.com/problems/flood-fill) | **Easy** | 20 mins |
| [Maximum Subarray](https://leetcode.com/problems/maximum-subarray) | **Easy** | 20 mins |
| [Lowest Common Ancestor of a Binary Search Tree](https://leetcode.com/problems/lowest-common-ancestor-of-a-binary-search-tree) | **Easy** | 20 mins |
| [Balanced Binary Tree](https://leetcode.com/problems/balanced-binary-tree) | **Easy** | 15 mins |
| [Linked List Cycle](https://leetcode.com/problems/linked-list-cycle) | **Easy** | 20 mins |

**Week 6**

| **Problem** | **Difficulty** | **Duration** |
| --- | --- | --- |
| [Implement Queue using Stacks](https://leetcode.com/problems/implement-queue-using-stacks) | **Easy** | 20 mins |
| [First Bad Version](https://leetcode.com/problems/first-bad-version) | **Easy** | 20 mins |
| [Ransom Note](https://leetcode.com/problems/ransom-note) | **Easy** | 15 mins |
| [Climbing Stairs](https://leetcode.com/problems/climbing-stairs) | **Easy** | 20 mins |
| [Longest Palindrome](https://leetcode.com/problems/longest-palindrome) | **Easy** | 20 mins |
| [Reverse Linked List](https://leetcode.com/problems/reverse-linked-list) | **Easy** | 20 mins |
| [Majority Element](https://leetcode.com/problems/majority-element) | **Easy** | 20 mins |
| [Add Binary](https://leetcode.com/problems/add-binary) | **Easy** | 15 mins |
| [Diameter of Binary Tree](https://leetcode.com/problems/diameter-of-binary-tree) | **Easy** | 30 mins |
| [Middle of the Linked List](https://leetcode.com/problems/middle-of-the-linked-list) | **Easy** | 20 mins |
| [Maximum Depth of Binary Tree](https://leetcode.com/problems/maximum-depth-of-binary-tree) | **Easy** | 15 mins |
| [Contains Duplicate](https://leetcode.com/problems/contains-duplicate) | **Easy** | 15 mins |

**Week 7**

| **Problem** | **Difficulty** | **Duration** |
| --- | --- | --- |
| [Min Stack](https://leetcode.com/problems/min-stack) | **Medium** | 20 mins |
| [Insert Interval](https://leetcode.com/problems/insert-interval) | **Medium** | 25 mins |
| [01 Matrix](https://leetcode.com/problems/01-matrix) | **Medium** | 30 mins |
| [K Closest Points to Origin](https://leetcode.com/problems/k-closest-points-to-origin) | **Medium** | 30 mins |
| [Longest Substring Without Repeating Characters](https://leetcode.com/problems/longest-substring-without-repeating-characters) | **Medium** | 30 mins |
| [3Sum](https://leetcode.com/problems/3sum) | **Medium** | 30 mins |
| [Binary Tree Level Order Traversal](https://leetcode.com/problems/binary-tree-level-order-traversal) | **Medium** | 20 mins |
| [Clone Graph](https://leetcode.com/problems/clone-graph) | **Medium** | 25 mins |
| [Evaluate Reverse Polish Notation](https://leetcode.com/problems/evaluate-reverse-polish-notation) | **Medium** | 30 mins |

**Week 8**

| **Problem** | **Difficulty** | **Duration** |
| --- | --- | --- |
| [Course Schedule](https://leetcode.com/problems/course-schedule) | **Medium** | 30 mins |
| [Implement Trie (Prefix Tree)](https://leetcode.com/problems/implement-trie-prefix-tree) | **Medium** | 35 mins |
| [Coin Change](https://leetcode.com/problems/coin-change) (dynamic programming) | **Medium** | 25 mins |
| [Product of Array Except Self](https://leetcode.com/problems/product-of-array-except-self) | **Medium** | 30 mins |
| [Validate Binary Search Tree](https://leetcode.com/problems/validate-binary-search-tree) | **Medium** | 20 mins |
| [Number of Islands](https://leetcode.com/problems/number-of-islands) | **Medium** | 25 mins |
| [Rotting Oranges](https://leetcode.com/problems/rotting-oranges) | **Medium** | 30 mins |
| [Search in Rotated Sorted Array](https://leetcode.com/problems/search-in-rotated-sorted-array) | **Medium** | 30 mins |

**Week 9**

| **Problem** | **Difficulty** | **Duration** |
| --- | --- | --- |
| [Combination Sum](https://leetcode.com/problems/combination-sum) | **Medium** | 30 mins |
| [Permutations](https://leetcode.com/problems/permutations) | **Medium** | 30 mins |
| [Merge Intervals](https://leetcode.com/problems/merge-intervals) | **Medium** | 30 mins |
| [Lowest Common Ancestor of a Binary Tree](https://leetcode.com/problems/lowest-common-ancestor-of-a-binary-tree) | **Medium** | 25 mins |
| [Time Based Key-Value Store](https://leetcode.com/problems/time-based-key-value-store) | **Medium** | 35 mins |
| [Accounts Merge](https://leetcode.com/problems/accounts-merge) | **Medium** | 30 mins |
| [Sort Colors](https://leetcode.com/problems/sort-colors) | **Medium** | 25 mins |
| [Word Break](https://leetcode.com/problems/word-break) | **Medium** | 30 mins |

**Week 10**

| **Problem** | **Difficulty** | **Duration** |
| --- | --- | --- |
| [Partition Equal Subset Sum](https://leetcode.com/problems/partition-equal-subset-sum) | **Medium** | 30 mins |
| [String to Integer (atoi)](https://leetcode.com/problems/string-to-integer-atoi) | **Medium** | 25 mins |
| [Spiral Matrix](https://leetcode.com/problems/spiral-matrix) | **Medium** | 25 mins |
| [Subsets](https://leetcode.com/problems/subsets) | **Medium** | 30 mins |
| [Binary Tree Right Side View](https://leetcode.com/problems/binary-tree-right-side-view) | **Medium** | 20 mins |
| [Longest Palindromic Substring](https://leetcode.com/problems/longest-palindromic-substring) | **Medium** | 25 mins |
| [Unique Paths](https://leetcode.com/problems/unique-paths) | **Medium** | 20 mins |
| [Construct Binary Tree from Preorder and Inorder Traversal](https://leetcode.com/problems/construct-binary-tree-from-preorder-and-inorder-traversal) | **Medium** | 25 mins |
| [Container With Most Water](https://leetcode.com/problems/container-with-most-water) | **Medium** | 35 mins |

**Week 11**

| **Problem** | **Difficulty** | **Duration** |
| --- | --- | --- |
| [Letter Combinations of a Phone Number](https://leetcode.com/problems/letter-combinations-of-a-phone-number) | **Medium** | 30 mins |
| [Word Search](https://leetcode.com/problems/word-search) | **Medium** | 30 mins |
| [Find All Anagrams in a String](https://leetcode.com/problems/find-all-anagrams-in-a-string) | **Medium** | 30 mins |
| [Minimum Height Trees](https://leetcode.com/problems/minimum-height-trees) | **Medium** | 30 mins |
| [Task Scheduler](https://leetcode.com/problems/task-scheduler) | **Medium** | 35 mins |
| [LRU Cache](https://leetcode.com/problems/lru-cache) | **Medium** | 30 mins |
| [Kth Smallest Element in a BST](https://leetcode.com/problems/kth-smallest-element-in-a-bst) | **Medium** | 25 mins |
| [Minimum Window Substring](https://leetcode.com/problems/minimum-window-substring) | **Hard** | 30 mins |

**Week 12**

| **Problem** | **Difficulty** | **Duration** |
| --- | --- | --- |
| [Serialize and Deserialize Binary Tree](https://leetcode.com/problems/serialize-and-deserialize-binary-tree) | **Hard** | 40 mins |
| [Trapping Rain Water](https://leetcode.com/problems/trapping-rain-water) | **Hard** | 35 mins |
| [Find Median from Data Stream](https://leetcode.com/problems/find-median-from-data-stream) | **Hard** | 30 mins |
| [Word Ladder](https://leetcode.com/problems/word-ladder) | **Hard** | 45 mins |
| [Basic Calculator](https://leetcode.com/problems/basic-calculator) | **Hard** | 40 mins |
| [Maximum Profit in Job Scheduling](https://leetcode.com/problems/maximum-profit-in-job-scheduling) | **Hard** | 45 mins |
| [Merge k Sorted Lists](https://leetcode.com/problems/merge-k-sorted-lists) | **Hard** | 30 mins |
| [Largest Rectangle in Histogram](https://leetcode.com/problems/largest-rectangle-in-histogram) | **Hard** | 35 mins |

**Factor time for your self introduction, final questions and mock coding interviews**[**​**](https://www.techinterviewhandbook.org/coding-interview-study-plan/#factor-time-for-your-self-introduction-final-questions-and-mock-coding-interviews)

Besides studying and practicing for coding interviews, you should also prepare your self introduction, final questions, and try out mock coding interviews.

**Prepare self introduction and final questions to ask**[**​**](https://www.techinterviewhandbook.org/coding-interview-study-plan/#prepare-self-introduction-and-final-questions-to-ask)

I would suggest around 3 hours to craft your self introduction and also prepare some final questions to ask. You may refer to this [self introduction guide](https://www.techinterviewhandbook.org/self-introduction/) and [final questions to ask guide](https://www.techinterviewhandbook.org/final-questions/) which should help you complete these steps fairly quickly.

**Schedule mock coding interviews**[**​**](https://www.techinterviewhandbook.org/coding-interview-study-plan/#schedule-mock-coding-interviews)

You should start scheduling for mock coding interviews when you are 60% through your coding interview studying and practicing plan. Interview slots are typically provided by interviewers, so you can view them in advance and book them. The platform I have personally used and recommend is [interviewing.io](https://iio.sh/r/DMCa). Read more about [different mock coding interview platforms here](https://www.techinterviewhandbook.org/mock-interviews/).

*Last updated on****Jul 3, 2022***